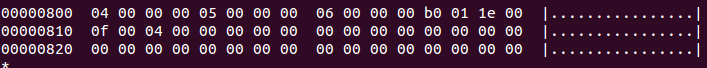
The dumpe2fs command gives us some help reading the hexdump. Each block is 1024 bytes and each inode is 128 bytes.

First, the superblock. Since a block is 1024 bytes, 400 in hex. The superblock starts at 400 and ends at 800.



At the line marked by 430, we see the magic number EF53 at the 9th block over. The 40 at byte 400 signifies that there are 64 inodes (40 in hex).

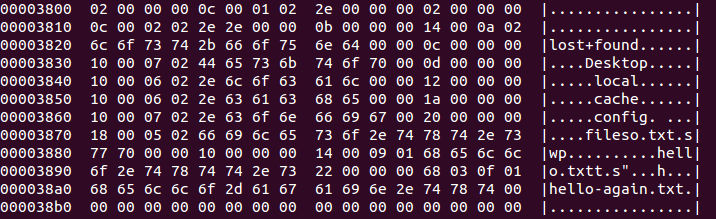
Next we have the files, directories and links. The inodes start at (6 \* 400) = 1800 as shown by the group descriptor.



1880 is the root node, since the root is inode 2. The i\_block shows that the directory entry for the root is at block 0e (byte 3800).



Here we have the directory entry for the root.

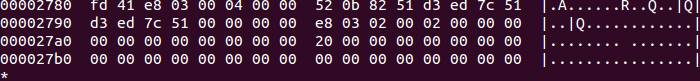


This shows that the files directory is in the 32nd inode (find the first character of the file name at the fifth btye at line 3870 and work backwards). This translates to (80 \* 20 -1) + 1800 = 2780.

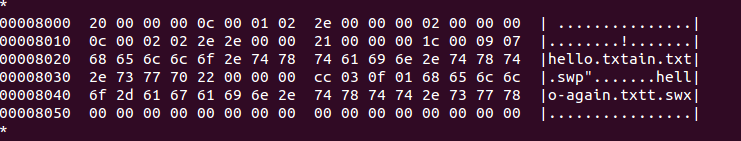
The file hello.txt is at the 16th inode.

The hard link to the file hello-again.txt is at inode 34.

Here is the inode for the files directory.

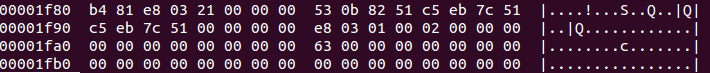


This shows that the directory is at block 20 (byte 8000).

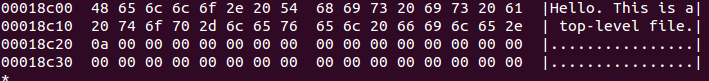


This shows the file hello.txt symbolic link is at the 33rd inode and the file hello-again.txt is at the 34th inode.

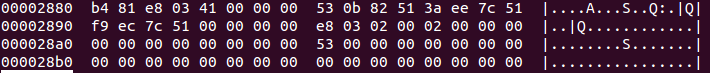
Here is the inode for the file hello.txt at byte (80 \* (10 – 1)) + 1800 = 1F80



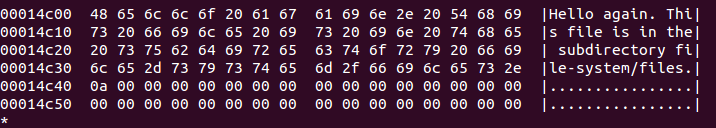
This shows the file is at block 63 or byte 18C00.



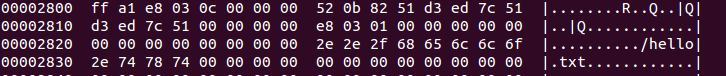
Here is the inode for the hard link to hello-again.txt at byte 2880. Note that the hard link and the actual file share an inode and a block.



This shows the file is located at block 53 or byte 14C00



Here is the inode for the symbolic link to hello.txt, which has a different inode than the original file, at byte 2800.



Rather than storing the location of the data block, it stores the path of the link, as seen by the /hello.txt in the canonical representation.